Warsaw University of Technology

Numerical Methods

Cholesky Method for solving linear equations

Authors:
Hassan Jaber

June 7, 2023

Contents

List of Figures

1 Project description

This is an implementation of a program that is written in matlab, that builds and tests the Cholesky method for solving linear equations: **Ax=b**, **where:** $A \in R^{n \times n}$ and $b \in R^n$, the determinant of the coefficient matrix A is calculated as well.

1.1 Requirements

1. The coefficient matrix A must be symmetric, positive definite, and pentadiagonal.
2. A and L must not be stored in square matrix, instead A is stored using 5 vectors, and L is stored using 3 vectors.

2 Theoretical description

2.1 Cholesky method

The Cholesky decomposition is mainly used for the numerical solution of linear equations $\mathbf{Ax} = \mathbf{b}$. If $\mathbf{A}$ is symmetric and positive definite, then we can solve $\mathbf{Ax} = \mathbf{b}$ by first computing the Cholesky decomposition $A = LL^t$, then solving $Ly = b$ for $\mathbf{y}$ by forward substitution, and finally solving $L^t x = y$ for $\mathbf{x}$ by back substitution.

Definition:

Let $L \in R^{n \times n}$ be a lower triangular matrix, that is $L_{ij} = 0$ if $i < j$, then :

$$\begin{bmatrix} \mathrm{L}_{11} & 0 & 0 & ... & 0 \\ \mathrm{L}_{21} & \mathrm{L}_{22} & 0 & ... & 0 \\ \mathrm{L}_{31} & \mathrm{L}_{32} & \mathrm{L}_{33} & ... & 0 \\ ... & ... & ... & ... & 0 \\ \mathrm{L}_{n1} & \mathrm{L}_{n2} & \mathrm{L}_{n3} & ... & \mathrm{L}_{nn} \end{bmatrix}$$

Let $L^t \in R^{n \times n}$ be a upper triangular matrix, that is the transpose matrix of L

$$\begin{bmatrix} \mathrm{L}_{11} & \mathrm{L}_{21} & \mathrm{L}_{31} & ... & \mathrm{L}_{n1} \\ 0 & \mathrm{L}_{22} & \mathrm{L}_{32} & ... & \mathrm{L}_{n2} \\ 0 & 0 & \mathrm{L}_{33} & ... & \mathrm{L}_{n3} \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & \mathrm{L}_{nn} \end{bmatrix}$$

After applying Cholesky method: A is decomposed into A = $\mathrm{L}L^t$

Calculating L: Suppose A is a symmetric positive matrix whose elements are denoted by a_{ij} . Let L be a lower triangular matrix whose elements are denoted by l_{ij} . The Cholesky decomposition can be represented in the following form:

$$\begin{cases} \mathrm{l}_{11} = \sqrt{a_{11}} \\ \mathrm{l}_{j1} = a_{j1}/l_{11},\ j \in [2, n] \\ \\ \mathrm{l}_{ii} = \sqrt{a_{ii} - \sum\limits_{k=1}^{i-1} {l_{ik}}^2},\ i \in [2, n] \\ \\ \mathrm{l}_{ji} = \left(a_{ij} - \sum\limits_{k=1}^{i-1} l_{ik} l_{jk}\right)/l_{ii},\ i \in [2, n-1],\ j \in [i+1, n] \end{cases} \tag{1}$$

Qingwen (2022)

Calculating the elements of the three vector of L in our special case: Let L1 be the main diagonal vector of the L, L2 is the first lower diagonal vector of the matrix, L3 is the second lower diagonal of the matrix.
Let M be the main diagonal vector of the coefficient matrix, U1 is the the first upper diagonal vector of the matrix, U2 is the second upper diagonal vector of the matrix. Let n be the size of the matrix.

$$\begin{cases} L1(1) = \sqrt{M(1)} \\ \\ L2(1) = \frac{U1(1)}{L1(1)} \\ \\ L1(2) = \sqrt{M(2) - (L2(1))^2} \\ \\ L3(1) = \frac{U2(1)}{L1(1)} \\ \\ L3(2) = \frac{U2(2)}{L1(2)} \\ \\ L2(2) = \frac{U1(2) - L2(1) \times L3(1)}{L1(2)} \\ \\ L1(3) = \sqrt{M(3) - (L3(1))^2 - (L2(2))^2} \end{cases} \tag{2}$$

$$\begin{cases} L3(j-2) = \frac{U2(j-2)}{L1(j-2)}, \quad j \in [4, n] \\ \\ L2(j-1) = \frac{U1(j-1)-L3(j-2)\times L2(j-2)}{L1(j-2)}, \quad j \in [4, n] \\ \\ L1(j) = \sqrt{M(j) - (L3(j-2))^2 - (L2(j-1))^2}, \quad j \in [4, n] \end{cases} \tag{3}$$

Calculating x:
After decomposing A into L and L^t, firstly we solve $Ly = b$ by forward substitution:

$Ly = b$ can be written as a system of linear equations:

$$\begin{cases} \mathrm{l}_{11}y_1 = b_1 \\ \mathrm{l}_{21}y_1 + l_{22}y_2 = b_2 \\ \ldots\ldots\ldots \\ \mathrm{l}_{n1}y_1 + l_{n2}y_2 + \ldots + l_{nn}y_n = b_n \end{cases} \tag{4}$$

The resulting formulas are:

$$\begin{cases} \mathrm{y}_1 = \frac{b_1}{l_{11}} \\ \\ \mathrm{y}_2 = \frac{b_2 - l_{21}y_1}{l_{22}} \\ \ldots \\ \mathrm{y}_n = \frac{b_n - \sum_{i=1}^{n-1} l_{ni}y_i}{l_{nn}} \end{cases} \tag{5}$$

Wikipedia (2022)

Now for solving $L^t x = y$ by back substitution, it can be done as same as foward substitution described above, only in a backwards way.

Calculating X in our special case:

Let L1 be the main diagonal vector of the L, L2 is the first lower diagonal vector of the matrix, L3 is the second lower diagonal of the matrix.
Let Y be the the solution vector for forward substitution.

Let n be the size of the matrix.

$$\begin{cases} Y(1) = b(1)/L1(1) \\ \\ Y(2) = \frac{b(2)-L2(1)\times y(1)}{L1(2)} \\ \\ Y(i) = \frac{b(i)-L2(i-1)\times y(i-1)-L3(i-2)\times y(i-2)}{L1(i)}, \quad i \in [3, n] \end{cases} \tag{6}$$

Now for solving $L^t x = y$ by back substitution, it can be done as same as foward substitution described above, only in a backwards way.

2.2 Symmetric positive definite pentadiagonal coefficient matrix:

Pentadiagonal matrix: Its only nonzero entries are on the main diagonal, and the first two upper and two lower diagonals. So it is of the form:

$$\begin{bmatrix} \mathrm{c}_1 & \mathrm{d}_1 & \mathrm{e}_1 & 0 & ... & ... & 0 \\ \mathrm{b}_1 & \mathrm{c}_2 & \mathrm{d}_2 & \mathrm{e}_2 & ... & ... & ... \\ \mathrm{a}_1 & \mathrm{b}_2 & ... & ... & ... & ... & ... \\ 0 & \mathrm{a}_2 & ... & ... & ... & \mathrm{e}_{n-3} & 0 \\ ... & ... & ... & ... & ... & \mathrm{d}_{n-2} & \mathrm{e}_{n-2} \\ ... & ... & ... & \mathrm{a}_{n-3} & \mathrm{b}_{n-2} & \mathrm{c}_{n-1} & \mathrm{d}_{n-1} \\ 0 & ... & ... & 0 & \mathrm{a}_{n-2} & \mathrm{b}_{n-1} & \mathrm{c}_n \end{bmatrix}$$

Symmetric Pentadiagonal matrix:

$$\begin{bmatrix} \mathrm{c}_1 & \mathrm{b}_1 & \mathrm{a}_1 & 0 & ... & ... & 0 \\ \mathrm{b}_1 & \mathrm{c}_2 & \mathrm{b}_2 & \mathrm{a}_2 & ... & ... & ... \\ \mathrm{a}_1 & \mathrm{b}_2 & ... & ... & ... & ... & ... \\ 0 & \mathrm{a}_2 & ... & ... & ... & \mathrm{a}_{n-3} & 0 \\ ... & ... & ... & ... & ... & \mathrm{b}_{n-2} & \mathrm{a}_{n-2} \\ ... & ... & ... & \mathrm{a}_{n-3} & \mathrm{b}_{n-2} & \mathrm{c}_{n-1} & \mathrm{b}_{n-1} \\ 0 & ... & ... & 0 & \mathrm{a}_{n-2} & \mathrm{b}_{n-1} & \mathrm{c}_n \end{bmatrix}$$

So as it is visible in the matrices above, whatever is the size of A, it can be stored in 5 vectors.

Positive Definite Definition for real matrices:
An $n \times n$ symmetric real matrix M is said to be **positive-definite** if x^TMx > 0 for all non-zero x in R^n.Formally,

M positive-definite $\iff x^T\text{Mx} > 0$ for all $x \in R^n \setminus \{0\}$

3 Program Description:

3.1 User Manual

User Input: In order to run this program, there is 6 inputs that should be initialized by the user, these 6 input represent the 5 vectors of the coefficient matrix in our linear equation (main diagonal, first upper diagonal, second upper diagonal,first lower diagonal, second lower diagonal), because in our case the coefficient matrix is pentadiagonal.

Since the requirement of the project are very specific about the coefficient (symmetric positive-definite pentadiagonal), then these vector are better not generated randomly by the program, they should be specified by the user, but in the testing program we generated these matrices for the testing sake.

The 6-th input represent b vector.

Program Output: The program output firstly represent the determinant of the coefficient, which is basically equal to the square of the determinant of L matrix since we are doing Chelosky decomposition:

$$det(A) = det(L)^2$$

Secondly, the other output is the vector x, which obviously represent the solution of the linear equation.

Example:

User Input:

```
upperdiag1 = [-1 -1 -1 -1 -1 -1 -1];
upperdiag2 = [-1 -1 -1 -1 -1 -1];
maindiag = [5 6 7 8 9 10 11 12];
lowerdiag1 = upperdiag1 // since it's symmetric
lowerdiag2 = upperdiag2
b = [1;2;3;4;5;6;7;8;]
```

Running the program:

```
[x, detA] = Cholesky(maindiag, upperdiag1, upperdiag2, lowerdiag1, lowerdiag2,
```

b)

Output:
```
x = 0.5192 0.7205 0.8756 0.9407 0.9651 0.9635 0.8922 0.8258

detA = 1.5216e+07
```

3.2 Cholesky method Algorithm

Here we are going to talk about, and show the algorithm of the program that consists of cholesky method, forward and back substitution for solving the linear equation, and the calculation of the determinant of A along the way.

3.2.1 Variables

Other than the variables that we talk about in the User Manual:

maindiag = a_{11}, a_{22}, ... ,a_{nn}
lowerdiag1 = a_{21}, a_{32}, ... ,$a_{n(n-1)}$
lowerdiag2 = a_{31}, a_{42}, ... ,$a_{n(n-2)}$
upperdiag1 = lowerdiag1
upperdiag2 = lowerdiag2
, there 4 more variables that are important that are initialized in the program itself, those variables are:
The three vectors **L1, L2, L3** that represent the matrix L:

L1= l_{11}, l_{22}, ... ,l_{nn}
L2 = l_{21}, l_{32}, ... ,$l_{n(n-1)}$
L3 = l_{31}, l_{42}, ... ,$l_{n(n-2)}$
The vector **y** that is used in both substitutions for solving the equation.

Algorithm 1 Cholesky Algorithm (Part 1)

function CHOLESKY(main_diag, upper_diag1, upper_diag2, lower_diag1, lower_diag2, b)
    $n \leftarrow \text{length}(\text{main\_diag})$
    $L1 \leftarrow \text{zeros}(n, 1)$
    $L2 \leftarrow \text{zeros}(n-1, 1)$
    $L3 \leftarrow \text{zeros}(n-2, 1)$
    $\text{detA} \leftarrow 1$
    $L1(1) \leftarrow \sqrt{\text{main\_diag}(1)}$
    $\text{detA} \leftarrow \text{detA} \times L1(1)$
    $L2(1) \leftarrow \frac{\text{upper\_diag1}(1)}{L1(1)}$
    $L1(2) \leftarrow \sqrt{\text{main\_diag}(2) - L2(1)^2}$
    $\text{detA} \leftarrow \text{detA} \times L1(2)$
    $L3(1) \leftarrow \frac{\text{upper\_diag2}(2)}{L1(1)}$
    $L3(2) \leftarrow \frac{\text{upper\_diag2}(2) - L1(1) \times 0}{L1(2)}$
    $L2(2) \leftarrow \frac{\text{upper\_diag1}(2) - L3(1) \times L3(1)}{L1(2)}$
    $L1(3) \leftarrow \sqrt{\text{main\_diag}(3) - L3(1)^2 - L2(2)^2}$
    $\text{detA} \leftarrow \text{detA} \times L1(3)$
    for $j = 4$ to n **do**
        $L3(j-2) \leftarrow \frac{\text{lower\_diag2}(j-2)}{L1(j-2)}$
        $L2(j-1) \leftarrow \frac{\text{lower\_diag1}(j-1) - L3(j-2) \times L2(j-2)}{L1(j-1)}$
        $L1(j) \leftarrow \sqrt{\text{main\_diag}(j) - L3(j-2)^2 - L2(j-1)^2}$
        $\text{detA} \leftarrow \text{detA} \times L1(j)$
    end for
    $\text{detA} \leftarrow \text{detA} \times \text{detA}$
    $y \leftarrow \text{zeros}(n, 1)$
    $y(1) \leftarrow \frac{b(1)}{L1(1)}$
    for $i = 2$ to n **do**
        if $i = 2$ **then**
            $y(i) \leftarrow \frac{b(i) - L2(i-1) \times y(i-1)}{L1(i)}$
        else
            $y(i) \leftarrow \frac{b(i) - L2(i-1) \times y(i-1) - L3(i-2) \times y(i-2)}{L1(i)}$
        end if
    end for

Algorithm 2 Cholesky Algorithm (Part 2)

34: $x \leftarrow \text{zeros}(n, 1)$
35: $x(n) \leftarrow \frac{y(n)}{L1(n)}$
36: **for** $i = n - 1$ to 1 step -1 **do**
37: **if** $i = n - 1$ **then**
38: $x(i) \leftarrow \frac{y(i)-L2(i)\times x(i+1)}{L1(i)}$
39: **else if** $i = n - 2$ **then**
40: $x(i) \leftarrow \frac{y(i)-L2(i)\times x(i+1)-L3(i)\times x(i+2)}{L1(i)}$
41: **else**
42: $x(i) \leftarrow \frac{y(i)-L2(i)\times x(i+1)-L3(i)\times x(i+2)}{L1(i)}$
43: **end if**
44: **end for**
45: **return** x, detA
46: **end function**

In the code above **(from line 7 to 23)**, it can be seen that we used equations**(1)** that we talked about earlier in order to calculate L matrix and its transpose, but we dealt with coefficient matrix A and L matrix as vectors. For solving the linear equation **(from line 25 to 44)** using forward and back substitution we used in the code the equations **(2)**,**(3)**. Finally we can see also, that the code update the determinant of A each time, an element of L1 is calculated, which represent the main diagonal of the matrix L, then at last we square it.

3.3 Testing Cholesky's Method

For testing the correctness of our implementation of Cholesky method, we implemented a program that would take one input n, which is the size of the coefficient that we desire, and the size of b vector.

The program would generate a random symmetric positive-definite pentadiagonal coefficient matrix of size n, and a random vector b as well.
In the testing program there is a loop that it will test if the generated matrix A is positive definite, if it's not, it will generate another one.

To insure that the matrix we are generating is symmetric pentadiagonal, the program actually generate a random matrix Q of size n, then we calculate the matrix P, which is equal to $Q^T \times Q$, then we take the main diagonal, two upper diagonals, and two lower diagonals. The resulting matrix is the testing coefficient matrix.

After generating the matrices and confirming that they are positive-definite, the program would solve the linear equation using the built in function in matlab **linsolve** to get the expected vector x, then it solve it again using our implementation of Cholesky method to get calculated vector x, finally the program will calculate the relative error using the following equation:

$$error1 = \frac{\|x - expected\_x\ \|}{\|expected\_x\|}$$

The program calculate other errors, like the forward stability error:

$$error2 = \frac{\|x - expected\_x\ \|}{\|expected\_x\| \cdot cond(A)}$$

Backward stability error:

$$error3 = \frac{\|b - A \cdot expected\_x\ \|}{\|A\| \cdot \|expected\_x\|}$$

Wrobel (2023)

Algorithm 3 Testing Cholesky method(part 1)

function TESTCHOLESKY(n)
    $isPositive \leftarrow$ false
    while ($isPositive ==$ false) **do**
        $Q \leftarrow \text{randi}([-10, 10], n)$
        $P \leftarrow Q' \cdot Q$
        $A \leftarrow \text{zeros}(n)$
        $A \leftarrow A + \text{diag}(\text{diag}(P)) + \text{diag}(\text{diag}(P, 1), 1) + \text{diag}(\text{diag}(P, 2), 2) + \text{diag}(\text{diag}(P, -1), -1) + \text{diag}(\text{diag}(P, -2), -2)$
        $[R, \text{flag}] \leftarrow \text{chol}(A)$
        if flag $== 0$ **then**
            $isPositive \leftarrow$ true
            $\text{disp}('Thematrixispositivedefinite.')$
        end if
    end while
    $b \leftarrow \text{randi}([1, 1000], n, 1)$
    tic()
    $[x, \text{detA}] \leftarrow \text{Cholesky}(\text{diag}(P), \text{diag}(P, 1), \text{diag}(P, 2), \text{diag}(P, -1), \text{diag}(P, -2), b)$
    elapsedTime $\leftarrow$ toc()
    $\text{cond\_A} \leftarrow \text{norm}(A) \times \text{norm}(\text{inv}(A))$
    $expected\_x \leftarrow \text{linsolve}(A, b)$
    $error1 \leftarrow \text{norm}(x - expected\_x)/\text{norm}(expected\_x)$
    $error2 \leftarrow \text{norm}(x - expected\_x)/(\text{norm}(expected\_x) \times \text{cond\_A})$
    $error3 \leftarrow \text{norm}(b - A \times expected\_x)/(\text{norm}(A) \times \text{norm}(expected\_x))$

Algorithm 4 Testing Cholesky method (Page 2)

print($ConditionalNumberOfA$)
print($RelativeError$)
print($ForwardError$)
print($BackwardError$)
print($ElapsedTime$)
end function

4 Results

In this section, we show the result of testing our implementation of Cholesky method:

Table 1 Result table

Size of matrix	Cond(A)	Relative Error	forward stability error	backward stability error	Execution time (seconds)
4	39.521138	6.6937e-16	1.694e-17	4.681e-17	3.355e-4
6	23.353786	2.4107e-16	1.032e-17	3.129e-17	2.251e-4
8	36.747967	9.7688e-16	2.658e-17	3.565e-17	2.201e-4
10	2598.559889	6.952e-14	2.676e-17	2.676e-17	3.573e-4
12	100.445501	2.3823e-16	2.37e-18	5.206e-17	5.778e-4
50	4.015552	1.6156e-16	4.023e-17	6.867e-17	4.98e-05
100	2.443150	1.4465e-16	5.921e-17	8.897e-17	4.74e-05
250	1.916334	1.5578e-16	8.129e-17	1.0920e-16	6.29e-05
500	1.554054	1.7174e-16	1.1051e-16	1.1434e-16	9.47e-05
1000	1.401542	1.4998e-16	1.0701e-16	1.1395e-16	1.018e-4
2500	1.259607	1.6948e-16	1.3455e-16	1.1858e-16	2.023e-4

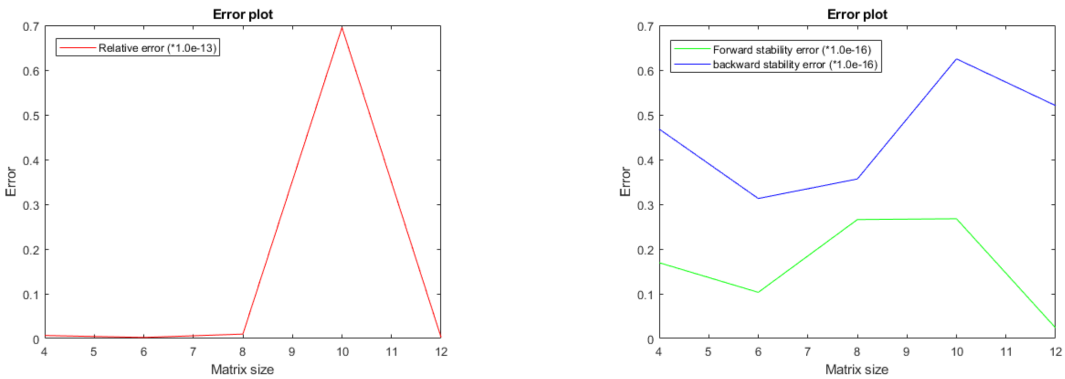

Figure 1: Errors plot

5 Conclusion

We managed to implement our own Cholesky method for solving linear equation, where the coefficient matrix is positive-definite, symmetric and pentadiagonal. And as we can see from the result table, the relative error, the execution time,forward stability error and backward stability error of our implementation are very low, even for ill-conditioned coefficient matrix, and for very large matrices. In conclusion, our implementation was pretty successful.

References

Qingwen, Liu (2022), "cholesky decomposition- university of illinois." URL https://faculty.math.illinois.edu/~psdey/MATH416SP22/reports/Liu%20Qingwen%20-%20CD.pdf.

Wikipedia (2022), "Wikipedia." URL https://en.wikipedia.org/wiki/Cholesky_decomposition.

Wrobel, Iwona (2023), "Lecture notes." URL https://pages.mini.pw.edu.pl/~wrobeli/WUTNM1_CSIS_summer_2022-23/.